METHOD FOR SEQUENCING POLYNUCLEOTIDES

FIELD OF THE INVENTION

This invention relates to computational methods in molecular biology, and more specifically to methods for determining the sequence of a polynucleotide.

REFERENCES

Baines, W., and Smith, G.C., *J. Theor. Biology*, **135**:303-307 (1988).

Ben-Dor, A., Pe'er, I., Shamir, R., and Sharan, R., In *Proceedings of the Tenth International Conference on Combinatorial Pattern Matching (CPM '99)*, 88-100 (1999), New York: ACM Press.

Cargill, M., Altshuler, D., Ireland, J., Sklar, P., Ardlie, K., Patil, N., Lane, C.R., Lim, E.P., Kalyanaraman, N., Nemesh, J., Ziaugra, L, Friedland, L., Rolfe, A., Warrington, J., Lipshutz, R., Daley, G.Q., and Lander, E.S., *Nature Genetics*, **22**:231-238 (1999).

Drmanac, R., and Crkvenjakov, R., Yugoslav Patent Application 570 (1987).

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G., *Biological Sequence Analysis: Probabilistic Models of proteins and Nucleic Acids*, Cambridge University Press, (1998).

Eddy, S.R., *Current Opinions in Structural Biology*, **6(3)**:361-365 (1996).

Hirschberg, D.S., *Communications of the ACM,* **18,6**:341-343 (1975).

Jukes, T.H., and Cantor, C.R., *Mammalian Protein Metabolism*, New York: Academic Press 21-123 (1969).

Khrapko, K.R., Lysov, Y.P., Khorlyn, A.A., Shick, V.V., Florentiev, V.L., and Mirzabekov, A.D., *FEBS Letters*, **256**:118-122 (1989).

Kimura, M., *Journal of Molecular Evolution*, **16**:111-120 (1980).

EXPRESS MAIL LABEL
NO.: EL699731101US

Krogh, A., Brown, M. Mian, S., Sjolander, M., and Haussler, D., Applications to protein modeling. Technical Report UCSC-CRL-93-32, Department of Computer and Information Sciences, University of California at Santa Cruz (1993).

Krogh, A., Brown, M., Mian S., Sjolander, M. and Haussler, D., Appliations to protein modeling **235(5)**:1501-1531 (1994).

Lysov, Y., Floretiev, V., Khorlyn, A., Khrapko, K., Shick, V., and Mirzabekov, A., *Dokl, Acad. Sci.*, USSR, **303**:1508-1511 (1988).

Macevices, S.C., International Patent Application PS US89 04741 (1989).

SUB B1

National Center for Biotechnology Information, 2000, A database of single nucleotide polymorphisms, http://www.ncbi.nlm.nih.gov/SNP/

Pevzner, P.A., and Lipshutz, R.J., *Mathematical Foundations of Computer Science*, LNCS **841**:143-158 (1994).

Pevzner, P.A., Lysov, Y. P., Khrapko, K.R., Belyavsky, A.V., Florentiev, V.L., and Mirzabekov, A.D., *J. Biomol. Struct. Dyn.* **7**:63-73 (1989).

Preparata, F., Frieze, A., and Upfal, E., *Journal of Computational Biology* **6**(3-4):361-368 (1999).

Skiena, S.S., and Sundaram, G., *J. Comput. Biol.* **2**:333-353 (1995).

Smith, T.F., and Waterman, M.S., *Journal of Molecular Biology* **147**(1):195-197 (1981).

Southern, E.M., Maskos, U., and Elder, J.K., *Genomics* **13**:1008-1017 (1992).

Southern E., UK patent Application GB 8,810,400 (1988).

Southern, E.M., *Trends in Genetics* **12**:110-115 (1996).

Wang, D.G., Fan, J., Siao, C., Berno, A., Young P., Sapolsky, R., Ghandour, G., Perkins, N., Winchester, E., Spencer, J., Kruglyak, L., Stein, L., Hsie, L., Topaloglou, T., Hubbell, E., Robinson, E., Mittmann, M., Morris, M.S., Shen, N., Kilburn, D., Rioux, J., Nusbaum, C., Lipshutz, R., Chee, M., and Lander, E.S., *Science* **280**:1077-1082.

Yang, Z., *Molecular Biology and Evolution* **10**:1396-1401.

BACKGROUND OF THE INVENTION

Sequencing by hybridization (SBH) is a method for sequencing a polynucleotide such as a DNA molecule (Bains & Smith 1988, Lysov *et al.* 1988, Southern 1988, Drmanac and Crkvenjakov 1987, Macevics 1989). In this method, a chip, or microarray is used consisting of a surface upon which all possible oligonucleotide probes of a particular length k (referred to herein as *"k-mers"*) are immobilized (Southern 1996). The DNA molecule whose sequence is to be determined, referred to as the *"target molecule"*, is allowed to hybridize to the k-mers on the chip. The target molecule and the k-mers on the chip may all be single stranded molecules. Alternatively, a double stranded target may first be cut into fragments having single stranded *"sticky ends"*, and the k-mers on the chip may be the sticky ends of double stranded molecules. Ideally, a single stranded target or the sticky end of a double stranded target hybridizes to a k-mer on the chip if and only if the sequence complementary to the k-mer occurs somewhere in the target sequence or the sticky end. Thus, in principle, it is possible to experimentally determine the *"k-spectrum"* of the target (the set of all k-long substrings present in the target). In practice, however, the data are ambiguous due to the ability of the target to bind to k-mers that are only partially complementary to one of its substrings. Thus, any binarization of the hybridization signal will contain errors.

The goal of SBH is to determine the target sequence from the target spectrum. However, even if the target spectrum were error free, the target sequence is not uniquely determined by the spectrum. If the number of sequences consistent with the spectrum is large, there is no satisfactory method to select the true sequence. Theoretical analysis and simulations (Southern *et al.*, 1992, Pevzner and Lipshutz 1994) have shown that even when the spectrum is errorless and the correct multiplicity of each k-mer in the target sequence is known, the average length of a uniquely reconstructible target sequence using a chip of 8-mers is only about two hundred nucleotides, far below the length of a DNA molecule that may be sequenced by electrophoresis.

Let Σ = (A,C,G,T) designate the set of nucleotides composing a DNA molecule. M = 4 is the *"alphabet size"*. A DNA sequence is a string over Σ which is denoted herein between braces (<>). The k-spectrum of a target sequence T of length L, T= <t_1, t_2,...t_L>, is the set of all k-long substrings (k-mers) of T. For each k-mer $\bar{x}$ = <x_1, x_2,...x_k> in $\in \Sigma^k$, we define $T(\bar{x})$to be 1 if $\bar{x}$ is a substring of T, and 0 otherwise. We denote K = M^k, the number of k-mers. A hybridization experiment measures, for each k-mer $\bar{x}$ in $\in \Sigma^k$, an intensity of its hybridization with the target.

The result of an SBH experiment may be described by a graph in which each candidate target sequence is represented as a path in a graph (Pevzner *et al.*, 1989). The graph is a directed de-Bruijn graph G(V,E) whose vertices are labeled by all the (k-1)-mers (the set of vertices V = Σ^{k-1}), and its edges are labeled by k-mers, (the set of edges E = Σ^k). The edge labeled <x_1, x_2...x_k> connects the vertex <x_1, x_2 ... x_{k-1}> to the vertex <x_2 ...x_k>. There is a 1-1correspondence between L-long candidate target sequences and (L - k + 1)- long paths in G, whose edge labels comprise the target spectrum. Hereafter, we interchangeably refer to edges and their labels, and also to sequences and their corresponding paths.

Since k-mers may reoccur in the target sequence, the paths do not have to be simple. When the spectrum is perfect and the multiplicities of the k-mers in the spectrum are known, every solution is an Eulerian path (Pevzner *et al.* 1989). In practice, however, the spectrum is not perfect and the multplicities are not known.

Alternative chip designs (Bains and Smith 1988, Khrapko *et al.* 1989, Pevzner *et al.* 1991, Preparata *et al.* 1999, Ben-Dor *et al.* 1999), as well as interactive protocols (Skiena and Sundaram 1995) have been suggested, often assuming additional information, in order to reduce the ambiguity of the hybridization-based reconstruction.

Nucleotide sequences from different sources may resemble each other, due to a common ancestral gene. This phenomenon is encountered within a species, between duplicated regions within a genome, and between individuals within a population. Small differences in sequences, referred to as *"Single Nucleotide Polymorphisms"* or *SNPs*, efficiently serve as genetic markers that are useful in medicine. Thus the detection and genotyping of SNPs has become an important task of human geneticists. The evolution of homologous sequences from a common ancestral gene is mainly due to nucleotide substitution. Insertions and deletions of nucleotides are also known to have occurred during evolution of homologous sequences, though at lower rates.

A DNA molecule having a known sequence and known to be homologous to a target molecule has not yet been used to reduce the ambiguity of SBH data in order to determine the target sequence.

SUMMARY OF THE INVENTION

In the following description and set of claims, two parameters are considered to be equivalent to each other if they are proportional to each other.

The present invention provides a method for sequencing a target sequence. In accordance with the invention, experimental spectrum data obtained from a DNA chip is combined with sequence information of a reference DNA molecule. The reference molecule is preferably a molecule believed to be homologous with the target. For example, the target sequence may be a mutant gene and the reference sequence the previously sequenced normal gene. As another example, the target sequence may be a human gene and the reference sequence the homologous gene in another organism. A score is defined for each sequence in a set of candidate target sequences based upon a simultaneous comparison of the candidate sequence with the spectrum and with the reference sequence. A candidate target sequence is then selected having a essentially maximal score. Calculating the score does not require knowledge of the multiplicities of the k-mers in the k-spectrum. Moreover, unlike all prior art algorithms, the score does not assume that the spectrum is perfect.

The invention therefore provides a novel probabilistic method that handles imperfect hybridization data with unknown multiplicities. Thus, in accordance with the invention the hybridization of the target T with the k-mer on the DNA chip complementary to $\vec{x}$ is described by probabilities $P_0(\vec{x})$ and $P_1(\vec{x})$ of the observed hybridization signal when $T(\vec{x}) = 0$, and $T(\vec{x}) = 1$, respectively The results of a hybridization experiment are described by the *"probabilistic spectrum"* (PS) defined as the pair (P_0,P_1) of functions P_i: $\Sigma^k \rightarrow [\,0,\ 1]$. If the experiment were perfect, i.e., if $P_0(\vec{x})$ and $P_1(\vec{x})$ are either 0 or 1 with $P_0(\vec{x})+P_1(\vec{x})\ \ 1$, then the PS would represent the k-spectrum. In practice, however, $P_0(\vec{x})$ and $P_1(\vec{x})$are both positive. There is thus a chance $1\text{-}P_0(\vec{x})$ for a false positive (a k-mer $(\vec{x})$ not occurring in T, whose complementary sequence produces a hybridization signal indicative of hybridization) and a chance $1\text{-}P_1(\vec{x})$ for a false negative (a k-mer $(\vec{x})$ occurring in T, whose complementary sequence produces a signal indicative of no hybridization). (When handling probabilities, some of which are perfect, problems of division by zero might occur. This is avoided by implicitly perturbing probabilities 0 and 1 to ε and $1\text{-}\varepsilon$.)

The probability of obtaining a specific spectrum PS when T is used as the target is referred to as the *"experimental likelihood"*. The experimental likelihood is calculated assuming that the hybridization results of the target to different k-mer probes are mutually independent. In one embodiment of the invention, an experimental likelihood $L^e(\hat{T})$ is used that does not assume knowledge of the multiplicities of each k-mer in the sequence. $L^e(\hat{T})$ is given by:

$$L^e\left(\hat{T}\right) = \mathrm{Prob}\left(PS \mid \hat{T}\right) = \prod_{\vec{x}\in\sum^k} P_{\hat{T}(\vec{x})}\left(\vec{x}\right) \qquad (1)$$

Taking logarithms and defining $\omega(\vec{x}) = log\frac{P_1(\vec{x})}{P_0(\vec{x})}$ we can write:

$$\log P_{\hat{\mathrm{T}}(\vec{x})}(\vec{x}) = \begin{cases} \log P_0(\vec{x}) & if \quad \hat{\mathrm{T}}(\vec{x})=0 \\ \log P_0(\vec{x})+\omega(\vec{x}) & if \quad \hat{\mathrm{T}}(\vec{x})=1. \end{cases} \qquad (2a)$$

Hence,

$$\log L^e(\hat{\mathrm{T}}) = \sum_{\vec{x}\in\sum^k} \log P_0(\vec{x}) + \sum_{\hat{\mathrm{T}}(\vec{x})=1} \omega(\vec{x}). \qquad (2b)$$

The first term is a constant (independent of $\hat{T}$), and is omitted hereafter.

In another embodiment, an approximate likelihood $\tilde{L}(\hat{\mathrm{T}})$ is used, that is defined as follows: Let p = e_0, …, $e_{L\text{-}k}$ be the path in G corresponding to $\hat{T}$ and define

$$log\,\tilde{L}^e(\hat{\mathrm{T}}) = \sum_{i=0}^{L-k} \omega(e_i) \,. \qquad (3)$$

$\tilde{\mathrm{L}}^e(\hat{\mathrm{T}})=\mathrm{L}^e(\hat{\mathrm{T}})$ for a path in which all edges have a multiplicity of 1, and is otherwise an approximation to $\mathrm{L}^e(\hat{\mathrm{T}})$. $\tilde{\mathrm{L}}^e(\hat{\mathrm{T}})$ has the advantage of being easily computable in a recursive manner:

$$log\,\tilde{L}^e(e_0,\ldots e_l) = log\,\tilde{L}^e(e_0,\ldots e_{l-1}) + \omega(e_l) \qquad (4)$$

In yet another embodiment, an experimental likelihood $\underline{\mathrm{L}}^e(\hat{\mathrm{T}})$ is used that takes into account the multiplicities of edges. In this case, the probabilistic spectrum consists of probabilities $\mathrm{P}_i(\vec{x})$, denoting the probability of the observed hybridization signal when the multiplicity of $\vec{x}$ in the target is i. $\underline{\mathrm{L}}^e(\hat{\mathrm{T}})$ is defined by:

$$\underline{L}^e(\hat{\mathrm{T}}) = Prob(PS \mid \hat{\mathrm{T}}) = \prod_{\vec{x}\in\sum^k} P_{\underline{\hat{\mathrm{T}}}(\vec{x})}(\vec{x}) \qquad (4b)$$

where $\underline{\hat{T}}(\vec{x})$ is the multiplicity of $\vec{x}$ in $\hat{T}$.

Thus in its first aspect, the invention provides a method for obtaining a candidate sequence, the candidate nucleotide sequence being indicative of a sequence of a target polynucleotide molecule T, T producing a hybridization signal I($\vec{x}$) upon incubating T with a polynucleotide $\vec{x}$ for each polynucleotide $\vec{x}$ in a set E of polynucleotides, the method comprising the steps of:

(a) for each polynucleotide $\vec{x}$ in the set E of polynucleotides, obtaining a probability $P_0(\vec{x})$ of the hybridization signal I($\vec{x}$) when the sequence $\vec{x}$ is not complementary to a subsequence of T and a probability $P_1(\vec{x})$ of the hybridization signal when the sequence $\vec{x}$ is complementary to a subsequence of T; so as to obtain a probabilistic spectrum (PS) of T;

(b) assigning a score to each of a plurality of candidate nucleotide sequences, the score being based upon the probabilistic spectrum and upon at least one reference nucleotide sequence H; and

(c) selecting one or more candidate nucleotide sequences having an essentially maximal score.

In its second aspect, the invention provides a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for obtaining a candidate nucleotide sequence, the candidate nucleotide sequence being indicative of a sequence of a target polynucleotide molecule T, T producing a hybridization signal I($\vec{x}$) upon incubating T with a polynucleotide $\vec{x}$ for each polynucleotide $\vec{x}$ in a set E of polynucleotides, the method comprising the steps of:

(a) for each polynucleotide $\vec{x}$ in the set E of polynucleotides, obtaining a probability $P_0(\vec{x})$ of I($\vec{x}$) when the sequence $\vec{x}$ is not complementary to a subsequence of T and a probability $P_1(\vec{x})$ of I($\vec{x}$) when the sequence $\vec{x}$ is complementary to a subsequence of T; so as to obtain a probabilistic spectrum (PS) of T;

(b) assigning a score to each of a plurality of candidate nucleotide sequences, the score being based upon the probabilistic spectrum and upon at least one reference nucleotide sequence H; and

(c) selecting a candidate nucleotide sequence having an essentially maximal score.

In its third aspect the invention provides a computer program product comprising a computer useable medium having computer readable program code embodied therein for obtaining a candidate nucleotide sequence, the candidate nucleotide sequence being indicative of a sequence of a target polynucleotide molecule T, T producing a hybridization signal $I(\vec{x})$ upon incubating T with a polynucleotide $\vec{x}$ for each polynucleotide $\vec{x}$ in a set E of polynucleotides, the computer program product comprising:

(a) for each polynucleotide $\vec{x}$ in the set E of polynucleotides, computer readable program code for causing the computer to obtain a probability $P_0(\vec{x})$ of $I(\vec{x})$ the sequence $\vec{x}$ is not complementary to a subsequence of T and a probability $P_1(\vec{x})$ of $I(\vec{x})$ when the sequence $\vec{x}$ is complementary to a subsequence of T;

(b) computer readable program code for causing the computer to assign a score to each of a plurality of candidate nucleotide sequences, the score being based upon the probabilistic spectrum and upon at least one reference nucleotide sequence H; and

(c) computer readable program code for causing the computer to select a candidate nucleotide sequence having an essentially maximal score.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

First Embodiment

In this embodiment, the unknown target sequence $T = <t_1 \ldots t_l>$ has a known, homologous reference sequence $H = <h_1 \ldots h_l>$. H and T are known to differ from each other by nucleotide substitutions without insertions or deletions (indels). This would be the case, for instance, when the target T is a mutant sequence whose wild type sequence *H* has already been sequenced, and one expects that nucleotide substitutions are the only cause of variability between *H* and T (statistically, substitutions are much more prevalent than indels (Wang *et* *al.* 1998)). A set of $M \times M$ position specific substitution matrices $M^{(1)}$, ..., $M^{(l)}$ are used, where for each position j along the sequence:

$$M^{(j)}[i,i'] = Prob\left(t_j = i \mid h_j = i'\right) \tag{5}$$

for nucleotides i and i'$\in\Sigma$.

The matrices $M^{(j)}$ may be the same for all j, or may different for different positions j. The matrices $M^{(j)}$ are used to calculate a distribution on the space of possible target sequences. This *"prior distribution for ungapped homology"*, D^u, is given, for each candidate target sequence T by:

$$D^u\left(\hat{T}\right) = Prob\left(\hat{T} \mid H\right) = \prod_{j=1}^{l} M^{(j)}\left[t_j, h_j\right] \tag{6}$$

One may recursively compute:

$$D^u\left(\left\langle t_1 \ldots t_j \right\rangle\right) = \left(\left\langle t_1 \ldots t_{j-1} \right\rangle\right) \cdot M^{(j)}\left[t_j, h_j\right] \tag{7}$$

We denote $L^{(j)}[x,y] \equiv \log M^{(j)}[x,y]$.

The probability of a candidate target sequence $\hat{T}$, given the probability spectrum PS and the reference sequence H is:

$$Prob\left(\hat{T} \mid H, PS\right) = \frac{Prob(H) \cdot Prob\left(\hat{T} \mid H\right) \cdot Prob\left(PS \mid H, \hat{T}\right)}{Prob(H, PS)} \tag{8}$$

Given $\hat{T}$, the hybridization signal is independent of *H*:

$$Prob\left(PS \mid H, \hat{T}\right) = Prob\left(PS \mid \hat{T}\right)$$

Thus, omitting the constant $\frac{Prob(H)}{Prob(H,PS)}$ we can write:

$$Prob\left(\hat{\mathrm{T}} \mid H,PS\right) \cong D^{u}\left(\hat{\mathrm{T}}\right) \cdot L^{e}\left(\hat{\mathrm{T}}\right) \tag{9a}$$

$$Prob\left(\hat{\mathrm{T}} \mid H,PS\right) \cong D^{u}\left(\hat{\mathrm{T}}\right) \cdot \tilde{L}^{e}\left(\hat{\mathrm{T}}\right) \tag{9b}$$

or

$$Prob\left(\hat{\mathrm{T}} \mid H,PS\right) \cong D^{u}\left(\hat{\mathrm{T}}\right) \cdot \underline{L}^{e}\left(\hat{\mathrm{T}}\right) \tag{9c}$$

Taking logarithms, the following *"ungapped scores"* of a candidate target are obtained:

$$Score_{1}^{u}\left(\hat{\mathrm{T}}\right) = \log L^{e}\left(\hat{\mathrm{T}}\right) + \log D^{u}\left(\hat{\mathrm{T}}\right) \tag{10a}$$

$$Score_{2}^{u}\left(\hat{\mathrm{T}}\right) = \log \tilde{L}^{e}\left(\hat{\mathrm{T}}\right) + \log D^{u}\left(\hat{\mathrm{T}}\right) \tag{10b}$$

$$Score_{3}^{u}\left(\hat{\mathrm{T}}\right) = \log \underline{L}^{e}\left(\hat{\mathrm{T}}\right) + \log D^{u}\left(\hat{\mathrm{T}}\right) \tag{10c}$$

With Score^{u}_{1} ,Score^{u}_{2} or Score^{u}_{3}, the higher the score of a sequence $\hat{T}$, the more likely it is to be the target sequence. The highest scoring candidate sequence may be determined by any method known in the art. In the search for the highest scoring candidate sequence, complexity is preferably reduced by deleting from the graph edges for which $\tilde{\mathrm{L}}^{e}\left(\hat{\mathrm{T}}\right)$ $\mathrm{L}^{e}\left(\hat{\mathrm{T}}\right)$ or $\underline{\mathrm{L}}^{e}\left(\hat{\mathrm{T}}\right)$ is less than a predetermined constant. Isolated vertices corresponding to highly improbable (k-1)-mers, are also preferably deleted from the graph.

For example, using $\tilde{\mathrm{L}}^{e}\left(\hat{\mathrm{T}}\right)$, the search for a high scoring candidate sequence may be performed by the following algorithm referred to herein as *"Algorithm A"*. In accordance with Algorithm A, for each vertex $\vec{y} = \langle y_{1} \ldots y_{k-1} \rangle \in \Sigma^{k-1}$, and integer $j = k-1, k, k+1, \ldots, l$, let $S^{u}[\vec{y}, j]$ be the maximum score of a j-long sequence ending with $\vec{y}$ aligned to $\langle h_{1} \ldots h_{j} \rangle$. Initialize, for each $\vec{y}$:

$$S^u[\vec{y},k-1]=\sum_{j=1}^{k-1}L^{(j)}[y_j,h_j] \qquad (11)$$

Loop over $j = k, \ldots, l$, and for each vertex $\vec{y}=\langle y_1 \cdots y_{k-1}\rangle$ recursively update:

$$S^u[\vec{y},j]=L^{(j)}[y_{k-1},h_j]+\max_{e=(\vec{z},\vec{y}\in E)}\left\{S^u[z,j-1]+\omega(e)\right\} \qquad (12a)$$

Finally, return:

$$MAX\,Score^u=\max_{\vec{y}\in V}S^u[\vec{y},l] \qquad (12b)$$

A sequence T* attaining the maximal score is found from the matrix S^u as is known in the art, for example, by saving trace-back pointers:

$$P[\vec{y},j]=\underset{\vec{Z}=<Z_0Z_1\ldots Z_{K-1}>,E=(\vec{Z},\vec{y})\in e}{\arg\max}\left\{S^u[z,j-1]+\omega(e)\right\} \qquad (13a)$$

$$MAXPtr=\underset{\vec{y}\in V}{\arg\max}\,S^u[\vec{y},l] \qquad (13b)$$

The maximum-scoring path in the graph is then followed, by setting: $Z^l=MAXPtr$, and for all $j = k, \ldots, l$: $Z^{j-1}=P[Z^j,j]$. Denote $Z^j=<z^j_1\, z^j_2 \ldots z^j_{k-1}>$. The final result is the sequence of nucleotides $<z^{k-1}_1,\ z^{k-1}_2, \ldots z^{k-1}_{k-1},\ z^k_{k-1},\ z^{k+1}_{k-1}, \ldots z^l_{k-1}>$

The time complexity is *O(lK)*, since the maximization in (12a),(13a) is a maximum of only a constant number (four) of terms. Although the complexity is exponential in k, it is constant for a given microarray (currently feasible values are k = 8 or 9). Moreover, the complexity scales linearly with the size of the hybridization experimental results, which are part of the input.

Space complexity requires a more elaborate analysis. When naively using this algorithm, it requires *O(lK)* memory space, which is quite high for current technology microarrays. We now detail how we can modify the algorithm to reduce space complexity.

Observe, that this algorithm consists of two computations: Computing the optimal score (equations (11),(12a) and (12b)), and reconstructing the optimal sequence (equations (13a) and (13b)). The first task, of computing the optimal score alone, is space-efficient: it can be accomplished using space which is linear in the (effective) size of the hybridization experimental data, that is, *O([K])* space.

By following the paradigm of Hirschberg (Hirschberg 1975), for example, for linear-space pair-wise alignment, a version of the algorithm is obtained which requires only linear space. The reduced space complexity is traded for time complexity, which increases by an *O(log l)* factor.

For each position $j = l, l\text{-}1, \ldots, k, k\text{-}1$, the score of the entire sequence is decomposed. The total score is represented as a sum of two expressions: the contribution of its $(j - k + 1)$-prefix, which equals the score of this prefix computed by S^u, plus the contribution of the corresponding suffix. Formally, for each vertex $\vec{y} = \langle y_1 \ldots y_{k-1} \rangle \in V$, let $R^u[\vec{y}, j]$ be the maximum contribution to the score of a $(l - j + k - 1)$-long sequence beginning with $\vec{y}$ aligned to $\langle h_{j-k+2} \ldots h_l \rangle$. Initialize, for each $\vec{y}$:

$$R^u[\vec{y}, l] = 0 \qquad (14)$$

Loop over $j = l - 1, l - 2, \ldots, k - 1$, and for each vertex $\vec{y} = \langle y_1 \ldots y_{k-1} \rangle$ recursively update:

$$R^u[\vec{y}, j] = \max_{e=(\vec{y},\vec{z}) \in E} \left\{ R^u[\vec{z}, j+1] + \omega(e) + L^{(j+1)}\left[z_{k-1}, h_{j+1}\right] \right\} \qquad (15)$$

Observe that, for all $k - 1 \leq j \leq l$

$$MAX\,Score^u = \max_{\vec{y}\in V}\left\{S^u[\vec{y}, j] + R^u[\vec{y}, j]\right\} \tag{16}$$

Equation (16) can be used to decompose the problem into two similar problems, of half its size. Recursively solving these sub-problems gives a divide-and-conquer approach for finding the optimal sequence. The linear space algorithm is therefore as follows:

1. If the length l of the target is smaller than some constant C, for example, 25 nucleotides:
   Solve the problem directly, according to the dynamic program of Equations (11), (12a), (12b), (13a) and (13b).

Otherwise,

2. Set $m = \frac{l + k - 1}{2}$.
3. For each $j = k - 1, \ldots, m$:
   Compute $S^u[\vec{y}, j]$ (following equations (11) and (12a)) for all $\vec{y}$, re-using space.
4. For each $j = l, l - 1, \ldots, m$:
   Compute $R^u[\vec{y}, j]$ (following equations (14) and (15)) for all $\vec{y}$, re-using space.
5. Find $\vec{y}_m = arg\,\underset{\vec{y}\in V}{max}\{S^u[\vec{y}, m] + R^u[\vec{y}, m]$,
   thereby computing: $MAX\,Score^u$, by (16).
6. Recursively compute:
   - (a) The optimal sequence aligned to $\langle h_1 \ldots h_m\rangle$ ending with $\vec{y}_m$.
   - (b) The optimal sequence aligned to $\langle h_m \ldots h_l\rangle$

beginning with $\vec{y}_m$.

Observe, that for each $\vec{y}$, j, the values of $S^u[\vec{y}, j]$ and $R^u[\vec{y}, j]$ are computed a total of log l times. Thus the algorithm takes *O([K]l* log *l*) time and *O([K]*) space, using the effective spectrum.

Second Embodiment: substitutions and deletions

In this embodiment, the unknown target sequence $\mathrm{T} = \langle t_1 \ldots t_{l''} \rangle$ differs from the reference $H = \langle h_1 \ldots h_l \rangle$, by substitutions and deletions only, without insertions.

Denote the probability of initiating a gap right before h_j (aligning h_j to *space*) is 2^{a_j}. Similarly, β_j is the logarithm of the probability for gap extension at h_j. Also define $\hat{\beta}_j = \log\left(1 - 2^{\beta_j}\right) \hat{\alpha}_j = \log\left(1 - 2^{\alpha_j}\right)$. To overcome boundary problems at the ends of the sequence, we extend the alphabet by including left and right space characters: $\overline{\Sigma} = \Sigma \cup \{\rhd, \lhd\}$. We augment the reference sequence by the string $\rhd^k$ on its left and $\lhd^k$ on the right. We extend the substitution matrix by using probabilities that force alignment of each of $\rhd$ and $\lhd$ to itself. Formally, we define:

$$\begin{aligned} \overline{\Sigma^{k-1}} = \Sigma^{k-1} \quad & \cup \quad \left\{ \vec{x}\vec{z} \mid \vec{x} = \rhd^{j}, \vec{z} \in \Sigma^{k-1-j} \right\} \\ & \cup \quad \left\{ \vec{z}\vec{x} \mid \vec{z} \in \Sigma^{j}, \vec{x} = \lhd^{k-1-j} \right\} \end{aligned} \tag{17}$$

We arbitrarily set $\omega(\vec{y})$ to 0 for each $\vec{y} \in \overline{\Sigma^{k-1}} \setminus \Sigma^{k-1}$. Thus, the weighted de-Bruijn graph is naturally extended over $\overline{\Sigma^{k-l}}$, and so is [G] = ([V], [E]), its effective subgraph. Hereafter, we use the notation [G] for the extended graph. As with the previous embodiment, in order to reduce complexity, edges for which $\tilde{\mathrm{L}}^{\mathrm{e}}(\hat{\mathrm{T}})$ or $\mathrm{L}^{\mathrm{e}}(\hat{\mathrm{T}})$ is less than ε are preferably deleted from the graph. Isolated vertices corresponding to highly improbable (k-1)-mers, are also preferably deleted from the graph.

The search for a high scoring candidate sequence may be performed by the following algorithm referred to herein as *"Algorithm B"*. In accordance with Algorithm B, for each $\vec{y}=\langle y_1 \dots y_{k-1}\rangle \in$ [V], $j = k = 1, k, k+1, \dots, l$, $S^d[\vec{y}, j]$ is defined as the maximum score of aligning a sequence ending with $\vec{y}$ to $\langle h_1 \dots h_j\rangle$ where h_j is aligned to a gap (and y_{k-1} is aligned to some $h_1 \dots h_j$). Further $T^d[\vec{x}, j]$ is defined as the maximum score of aligning a sequence ending with $\langle y_1 \dots y_{k-1}\rangle$, to $\langle h_1 \dots h_j\rangle$ where h_j aligned to y_{k-1}. Initialize, for each $\vec{y}$:

$$S^d[\vec{y}, k-1] = -\infty; \tag{19}$$

$$T^d[\vec{y}, k-1] = \begin{cases} 0 & \vec{y} = \triangleright^{k-1} \\ -\infty & otherwise \end{cases} \tag{20}$$

Loop over j = k, ..., l, and for each $\vec{y}=\langle y_1 \dots y_{k-1}\rangle \in$, [V], recursively update:

$$S^d[\vec{y}, j] = \max\{T^d[\vec{y}, j-1] + \alpha_j, S^d[\vec{y}, j-1] + \beta_j \tag{21}$$

$$T^d[\vec{y}, j] = L^{(j)}[y_{k-1}, h_j] + \max_{e=(\vec{z},\vec{y})\in E}\left\{\omega(e) + max\left\{\begin{matrix} T^d[\vec{z}, j-1] + \hat{\alpha}_j \\ S^d[\vec{z}, j-1] + \hat{\beta}_j \end{matrix}\right\}\right\} \tag{22}$$

Finally, return:

$$MAX\,Score^{d} = T^{d}\left[\lhd^{k-1}, l\right] \tag{23}$$

The complexity of this algorithm is still $O(l[K])$ and a linear space variant can be obtained, as described in the previous embodiment. A sequence T^* attaining the maximal score is then formed from the matrix T^d as is known in the art, for example, by saving trace-back pointers to follow the maximally scoring path in analogous manner to that described in the previous embodiment.

Third Embodiment: Substitutions, Deletions and Insertions.

In this embodiment, a target sequence is determined when the target is known to be obtained from the reference by substitutions, insertions and deletions. The algorithm is an extension of the dynamic programs of the previous embodiments.

Denote by T_j the target prefix whose last nucleotide is aligned to h_j in the reference sequence. Further denote by a_j (respectively b_j) the log-probability of initiating (extending) an insertion in the target after T_j, and define $\hat{a}_j = 1 - a_j, \hat{b}_j = 1 - b_j$.

Consider the weighted graph (G,ω). Define the K x K matrix W as follows:

$$W[\vec{x},\vec{y}] = \begin{cases} 2^{w(\vec{y})} & The\ (k-1)-suffix\ of\ \vec{x} \\ & is\ the\ (k-1)-prefix\ of\ \vec{y} \\ 0 & Otherwise \end{cases} \tag{24}$$

$W^i[\vec{x},\vec{y}]$ is thus the probability of moving from $\vec{x}$ to $\vec{y}$ along i edges. The probability of an insertion of length i after T_j is $a_j b_j^i \hat{b}_j$. Suppose that the prefix T_j ends with $\vec{x}$. Then $a_j b_j^{i-1} \hat{b}_j W^i[\vec{x},\vec{y}]$ is the probability of T_{j+1} ending with $\vec{y}$

and being i nucleotides longer than T_j. The matrix W' governing the stochastic progression from T_j to T_{j+1} is calculated as follows:

$$W' = \hat{a}_j b_j W^2 \hat{b}_j + a_j b_j^2 W^3 \hat{b}_j \ldots \tag{25}$$

$$= \hat{a}_j W + a_j b_j \hat{b}_j W^2 \sum_{i \geq 2} b_j^{i-2} W^{i-2} \tag{26}$$

$$= \hat{a}_j W + a_j b_j \hat{b}_j W^2 \left(I - b_j W \right)^{-1} \tag{27}$$

A new weighted graph (G',ω') is now defined as follows. The vertex set of G is also the vertex set of G'. The edge set E' of G' is the set of all pairs $\vec{x}, \vec{y}$ with $W'[\vec{x},\vec{y}] \rangle 0$. Each such edge $e = [\vec{x},\vec{y}]$ is associated with a weight $w'(e)$ = log $W'[\vec{x},\vec{y}]$.

The search for a high scoring candidate sequence may be performed by the following algorithm referred to herein as *"Algorithm C"*. In accordance with Algorithm C, Algorithm B of the second embodiment is applied to (G',ω') instead of (G, ω).

In contrast to G, degrees in G' are not bounded by 4. Therefore, computing each dynamic program cell has complexity $O(K)$ in the worst case, with the total complexity of the algorithm being $O(l|E'|)$. Again, considering only the effective size of the graph allows more efficient computation, taking $O(l|[E']|)$.

Fourth Embodiment: Substitutions, Deletions and Insertions.

In this embodiment, homology between nucleotide sequences is described by Hidden Markov Models (HMMs) using a set Q of Markov chain states with a predefined set of allowed transitions between them. For each level (position along the sequence) j = 1, ..., l_Q, Q includes three states: M_j (match), I_j (insert), and D_j (delete). M_j and D_j can be reached from the three (j-1) (th) level states. I_j

can be reached from the three (j)-(th) level states (including a self-loop). Transition probabilities are as described in previous sections, e.g., a_j = $Prob(M_j \mapsto I_j)$. Additionally, each insert or match state, q, induces a vector of emission probabilities M^q, where $M^q[i]$ is the probability that the target nucleotide is i. We denote $L^q[i] \equiv 0$ for $q = D_j$, $L^q[i] \equiv \log M^q[i]$ otherwise. We write $lpb(X) \equiv$ log $Prob(X)$ for short.

The search for a high scoring candidate sequence may be performed by the following algorithm referred to herein as *"Algorithm D"*. In accordance with Algorithm D, a three dimensional array S is defined, where for each $q \in Q, \vec{y} = \langle y_1 \ldots y_{k-1} \rangle \in [V], r = k, \ldots,\ L,\ S[q, \vec{y}, r]$ is defined as the maximum score of an r-long sequence ending with $\langle y_1 \ldots y_{k-1} \rangle$, whose alignment to the profile ends in q. Thus, initialize:

$$S\left[q_{start.}\ \rhd^{k-1}, k-1\right] = 0 \tag{28}$$

$$S[q, \vec{y}, k-1] = -\infty \quad \textit{for other values of } \vec{y}, q \tag{29}$$

Loop over r = k, ... l, and for each $\vec{y} = \langle y_1 \ldots y_{k-1} \rangle \in [\mathrm{V}], r \le l_Q$, recursively update:

$$S[q, \vec{y}, r] = L^q[y_{k-1}] + \max_{\substack{e=(E,q)\in E \\ q'|q' \mapsto q}} \{S[q', \vec{z}, r-1] + lpb(q' \mapsto q) + \omega(e)\} \tag{30}$$

Finally, return:

$$MAX\ Score = \max_l \left\{ S\left[q_{end1}\ \lhd^{k-1}, l\right] \right\} \tag{31}$$

A sequence T* is maximal score is then found in a manner similar to that described in the previous embodiments.

This algorithm requires $O(l_Q \cdot [K] \cdot L)$ time and space, where L is an upper bound on the size of the target sequence. As with the previous embodiments, the complexity of this algorithm can be reduced to $O(l_Q \cdot [K] \cdot L \log L)$ time and $O(l_Q \cdot [K])$ memory. Furthermore, one can consider the dynamic program as filling a l_Q x L matrix, with a $[K]$-long vector in each matrix cell. Since all values far from the main diagonal of this matrix should be negligible, preferably only values within a distance less than a predetermined constant R to the main diagonal are calculated, reducing the complexity to $O(R(l_Q+L) \cdot [K] \cdot \log L)$ time and $O(R(l_Q+L) \cdot [K])$ space.

Fifth Embodiment: Summation over all paths

In this embodiment the graph nodes (HMM states and k-mers) that are most likely to be visited at a certain position along the target sequence are obtained. The *"Forward-Backward"* algorithm is used (see, e.g., Durbin et al., 1998) providing the likelihood summed over all paths entering a node, instead of the likelihood of the maximum path. The only change to the equation presented thus far is that *max* operators are changed into *log-sum-of-exponents*. More specifically, equations (12a), (12b), (15), (16), (20), (21), (29), and (30) are re-written, respectively, as follows:

$$S^u[\vec{y}, j] = L^{(j)}[y_{k-1}, h_j] + \log \sum_{e=(\vec{z},\vec{y}\in E)} \exp(S^u[z, j-1] + \omega(e)) \quad (12a')$$

$$MAX\ Score^u = \log \sum_{\vec{y}\in V} \exp(S^u[\vec{y}, l]) \quad (12b')$$

$$R^u[\vec{y}, j] = \log \sum_{e=(\vec{y},\vec{z})\in E} \exp(R^u[\vec{z}, j+1] + \omega(e) + L^{(j+1)}[z_{k-1}, h_{j+1}]) \quad (15')$$

$$MAX\,Score^u = \log \sum_{\vec{y}\in V} \exp(S^u[\vec{y}, j] + R^u[\vec{y}, j]) \quad (16')$$

$$S^d[\vec{y},j]=\log\left(\exp\left(T^d[\vec{y},j-1]+\alpha_j\right)+\exp\left(S^d[\vec{y},j-1]+\beta_j\right)\right) \qquad (20')$$

$$T^d[\vec{y},j]=\begin{array}{l} L^{(j)}\left[y_{k-1},h_j\right]+\log\sum_{e=(\vec{z},\vec{y})\in \mathrm{E}}\exp(\omega(e))+ \\ +\log\left(\exp\left(T^d[\vec{z},j-1]+\hat{\alpha}_j\right)+\exp\left(S^d[\vec{z},j-1]+\hat{\beta}_j\right)\right)\end{array} \qquad (21')$$

$$S[q,\vec{y},r]=L^q\left[y_{k-1}\right]+\sum_{\substack{e=(E,q)\in E \\ q'|q'\mapsto q}}\exp\left(S[q',\vec{z},r-1]+lpb(q'\mapsto q)+\omega(e)\right) \qquad (29')$$

$$MAX\ Score=\log\sum_{l}\exp\left(S\left[q_{end1}\ \triangleleft^{k-1},l\right]\right) \qquad (30')$$

Sixth Embodiment: Enhancements

In this embodiment the exact likelihood calculated according to Equation 10a of several top-scoring candidates found using the approximated likelihood (Equation 10b) is calculated. These sequences are then re-ranked. This 2-phase filtering is more discriminative than approximated scoring, while still tractable using the formulae presented.

If the score of a dynamic programming cell is very low, that cell probably does not participate in the maximum solution. This allows discarding such cells, without risking loss of the true optimum. Computing time and space may thus be saved.

The invention may be used for simultaneously re-sequencing several short targets, instead of a single long sequence. This scenario arises when considering many exons of a single gene. The invention may also be generalized to deal with DNA chips that do not contain the set of all k-mers.

When the set of oligonucleotides on the microarray is not the set of all k-mers, a graph is constructed consisting, as vertices, instead of all the (k-1)-mers, all the prefixes and suffixes of oligonucleotides on the microarray. Edges in this graph connect two vertices if there is one base pair suffix (suffix) addition to one of them, that makes the other its proper suffix (prefix). The scoring mechanism remains the same. This also applies for oligonucleotides containing “gaps” or “universal bases” (Preparata et al., 1999).

The invention may be used also for sequencing polypeptides. Given a polypeptide chain homologous to a target, and given a collection of probabilities of occurrence of sub-chains along the target, our algorithms will find the optimal candidate target sequence.

Example

The invention was implemented and tested on simulated data. Nucleotide substitutions were equiprobable and insertions and deletions were not allowed. As a reference sequence, prefixes of the gene-rich human mitochondrial sequence, (Accession Number J01415) were used. For each reference sequence, the following were generated:

1. A target sequence generated according to a prescribed probability q of substitution, defining the matrix M as 1-q on the diagonal and q/3 elsewhere.
2. An 8-spectrum for the target was generated using the probabilistic spectrum defined by $P_i(\vec{x}) = 1 - p$ if $\mathrm{T}(\vec{x}) = i$, where p is a fixed probability.

All probabilistic parameters were constant, i.e., position/*k*-mer independent. For each 8-spectrum and target sequence, candidate sequences were scored using Eq. (10), and a candidate sequence of maximal score was found.

The algorithm was implemented in *C*++ and executed on Linux and SGI machines. Running times, on a Pentium 3, 600MHz machine, were roughly 0.12*l* log *l* seconds for an *l*-long reference sequence (ranging from roughly 7 minutes for a 500bp-long sequence to 2.5 hours for 6Kb). Only the main memory was used, with the application consuming at most 40Mb.The graph was not reduced to its effective size. This would have reduced both space and time dramatically, at the expense of possibly missing the truly maximal scoring sequence.

The performance of the algorithm was quantified by the following figures of merit:

1. Full success rate-The fraction of runs for which the target sequence was perfectly reconstructed.

2. ε-success rate - The fraction of runs for which the target sequence of length l was reconstructed with fewer than $\varepsilon \bullet l$ nucleotide errors.

3. Average sequencing error - The fraction of nucleotide errors.

Table 1 presents results for a scenario of distinct, but closely related sequences, e.g., orthologous genes in a pair of primates. We assume perfect hybridization data with 97% sequence similarity (that is q=0.03). The results show that sequences of length up to 2000 can be reconstructed almost perfectly. The non-monotonicity of the figures of merit with respect to the target length is probably due to sequence contents.

Table 2 presents results for a scenario of SNP-genotyping. The rate of SNPs is assumed to be 1:700 (Wang *et al.* 1998), and p=2% was used. The results show that a high success rate is achievable even in the presence of spectrum errors.

Table 1

Length	# runs	% full success	% ε-success		% avg. error
			$\varepsilon = 10^{-3}$	$\varepsilon = 2 \cdot 10^{-3}$	
500	10	100	100	100	0.000
1000	10	100	100	100	0.000
1500	10	100	100	100	0.000
2000	17	94	94	94	0.003
2500	13	46	53	69	0.295
3000	14	71	78	78	0.488
3500	5	0	20	20	4.091
4000	13	76	84	84	2.173
4500	11	9	18	45	0.091
5000	15	0	13	53	4.149
5500	7	14	28	71	0.119

Table 1: Results on simulated date, for different sequence lengths, assuming 97% sequence similarity between the target and the reference, and perfect hybridization data.

Table 2

Length	# runs	% full success	% ε-success		% avg. error
			$\varepsilon = 10^{-3}$	$\varepsilon = 2{\cdot}10^{-3}$	
250	10	100	100	100	0.000
500	10	100	100	100	0.000
750	10	90	90	100	0.013
1000	10	90	90	90	0.010
1250	10	90	100	100	0.032
1500	12	91	100	100	0.033
1750	10	60	80	80	0.109
2000	10	60	90	90	4.965
2500	10	0	80	100	10.312
3000	10	30	70	90	0.230

Table 2: Results on simulated data, for different sequence lengths, assuming $p = 2\%$ error the hybridization data, with 1:700 sequence difference.

It will also be understood that the system according to the invention may be a suitably programmed computer. Likewise, the invention contemplates a computer program being readable by a computer for executing the method of the invention. The invention further contemplates a machine-readable memory tangibly embodying a program of instructions executable by the machine for executing the method of the invention.